REMARKS

In this Office Action, the Examiner objected to the ABSTRACT of the Invention. The Examiner also objected to Claims 1, 2, 7 – 9, 18 – 20, 26, 27, 32 – 34, 43, 44 and 51 because of some informalities. Claims 6, 21, 31 and 45 were rejected under 35 U.S.C. §112, first paragraph, as failing to comply with the enablement requirement. Claims 1 – 11, 13 – 22, 25 – 36, 38 – 46 and 48 – 53 were rejected under 35 U.S.C. §102(b) as being anticipated by Kamdar. Claims 11, 12, 23, 24, 36, 37 and 47 were objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form to include all of the limitations of the base claim and any intervening claims.

The Examiner is thanked for the interview of February 15, 2006. In that interview, Applicants' attorney, Examiner Thompson and Examiner Rossochek discussed Claim 1 and the applied reference. Specifically, Applicants' attorney explained that Claim1 is amended to specify that the "graph is made of a plurality of subgraphs that include a plurality of elements and that when elements of a subgraph are repositioned other subgraphs, if affected, will be repositioned (i.e., all the elements that make up an affected subgraph will be repositioned as a group)." The Examiner mentioned that she will have to undertake another search to determine allowability.

The Examiner also pointed out that the following limitations in the amended Claim 1 were still not definitive: "each of said plurality of subgraph classes ~~adapted~~ being enabled to:". Applicants' attorney agrees to further amend the claims for definitiveness.

The Examiner rejected Claims 11 and 36 under 35 U.S.C. §102(b) and yet indicated that they were allowable if rewritten in independent form to include all of the limitations of the base claim and any intervening claims. In this Response, the two claims are treated as being allowable since they contain the limitations, which the Examiner indicated, distinguished the (other) allowable claims from the applied reference. However, if the claims are indeed rejected under 35 U.S.C.

§102(b), Applicants respectfully request that the Examiner do indicate so in the next Office Action.

In response to the objection to the ABSTRACT, Applicants have provided a new ABSTRACT to replace the offensive ABSTRACT.

In addition, Applicants have amended Claims 1, 2, 7 – 9, 19, 20, 26, 27, 32, 33, 43, 44 and 51 to overcome the 112 objection made thereto. Claims 6, 21 and 31 were amended and Claim 45 canceled to overcome the 112, first paragraph, rejections.

Applicants have also amended independent Claims 1, 2, 6 – 9, 15, 19, 20 – 23, 26, 27, 31 – 33, 37, 43 44, 49 – 51 and canceled Claims 5, 18, 30, 34, 36, 38, 39, 45 – 48 in order to better claim the invention as well as to provide definiteness to the claimed invention. No new matter has been added.

By this amendment, Claims 1 – 4, 6 – 17, 19 – 29, 31 – 33, 35, 37, 40 – 44 and 49 - 53 remain pending in the Application. For the reasons stated more fully below, Applicants submit that the claims are allowable over the applied reference. Hence, reconsideration, allowance and passage to issue are respectfully requested.

As mentioned in the SPECIFICATION, constrained graphs are graphs constructed using graphical elements that have been predefined in how they are composed and sequenced, which may be used to build complex graphical flows and which may be displayed in software applications. Graphical flows can be used in the modeling of various processes.

In a layout of the graphical elements of a constrained graph, there are certain rules, defined at development time which the layout must conform to. Accordingly, the term "constrained" is used in reference to the graphical elements available in building the graph and to the rules on how these graphical elements may be combined.

The present invention provides means to dynamically manage an expanding and contracting graph in response to user interactions such as insertions and deletions of graphical elements in the graph. To make a large,

complex, constrained graph more manageable, in accordance with the present invention, the constrained graph is broken down into smaller parts, referred to as subgraphs. Each subgraph is also composed of one or more graphical elements (e.g. nodes, terminals, connections, bundles of connections, other subgraphs). However, a subgraph is not merely an arbitrary collection of graphical elements. Each type of subgraph is characterized by a specific grouping or combination of graphical elements, which is defined by the software developer.

When a user deletes a graphical element from a subgraph or adds a graphical element to a subgraph, the graphical elements of the subgraph are repositioned. If other subgraphs are affected by the repositioning of the graphical elements of the subgraph, the other subgraphs are also repositioned.

The invention is set forth in claims of varying scopes of which Claim 1 is illustrative.

> 1. A software system for constrained graphs comprising:
>
> software code for implementing a graph, ***said graph being constructed using a plurality of subgraphs having each a pre-defined grouping of a plurality of graphical elements***;
>
> software code for repositioning elements of a subgraph; and
>
> ***software code for repositioning other subgraphs when the other subgraphs are affected by the repositioning of the elements of the subgraph***. (Emphasis added.)

The Examiner rejected the claims under 35 U.S.C. §102(b) as being anticipated by Kamdar. Applicants respectfully disagree.

Kamdar purports to teach a method and apparatus for constraining the compaction of components of a circuit layout. According to the purported teachings of Kamdar, integrated circuits (ICs) may be designed using a computer program known as IC design tool. With this tool, a circuit designer enters symbolic or geometrical shapes representing elements of an IC design into a

computer and manipulates the size and location of the elements to produce a simulated layout of the circuit. From this simulated layout, the design tool generates data for producing photolithographic masks that are then used for fabricating the IC.

Manipulating the size and location of the elements to produce a simulated layout of the circuit can be a rather tedious task. Thus, some IC design tools include a compactor that speeds layout design by automating the task of reducing the design area. A user gives the compactor a preliminary layout. The compactor then moves circuit elements, or components, of the design to optimize for two goals: that the layout be small and that it be design rule correct. The user can therefore have a great deal of control on the layout without performing the tedious work required to turn a sketch of a layout into a correct, space-optimized design.

In some instances, relative positions of certain components need to be preserved after compaction. This does not always happen, especially when device sizes change because of a change in process technology or a change in parameters. This can lead to unexpected and undesirable compaction results. Thus, Kamdar advocates the use of logical boundary constraints or pre-compaction constraints to assure that the relative positions of components in an initial layout are preserved after compaction.

In accordance with the purported teachings of Kamdar, each of the physical boundaries of a layout has a corresponding logical boundary. Hence, there are left, right, bottom, and top logical boundaries corresponding to the left, right, bottom and top physical boundaries. A logical boundary is in essence a "soft" physical boundary. By default, the logical boundaries coincide with their physical boundaries. However, when logical boundary constraints (LBCs) are specified by a user, the logical boundaries might be offset from their corresponding physical boundaries. This offset (if any) depends upon the layout components for which LBCs have been specified and the reference point on the

constrained component which is referred by the LBC, and can vary from one compaction pass to another.

In any event, physical boundary constraints force one of the edges of the constrained element to be necessarily aligned with the corresponding physical layout boundary. One important difference between a logical boundary constraint and a physical boundary constraint is that a logical boundary constraint may or may not cause the constrained component to have one of its edges aligned with a physical boundary as long as there is at least one other component with a logical boundary constraint which has an edge aligned with the corresponding edge of the layout. Irrespective of whether the component constrained by an LBC has an edge aligned with a layout edge or not, all the reference points of the components constrained by the LBCs to the same boundary are aligned during the compaction process. This ensures that their relative positions are preserved.

However, Kamdar does not teach, show or so much as suggest the limitations of ***implementing a graph that is constructed using a plurality of subgraphs having each a pre-defined grouping of a plurality of graphical elements***, and of ***repositioning other subgraphs when the other subgraphs are affected by the repositioning of the elements of the subgraph*** as claimed.

Consequently, Applicants submit that Claim 1, as well as its dependent claims should be allowable. The other independent claims (i.e., Claims 19, 26, 43, 44, 49 and 50), which all incorporate in one fashion or another the emboldened-italicized limitations in the above-reproduced Claim 1 and their dependent claims should be allowable as well. Consequently, Applicants once more respectfully request reconsideration, allowance and passage to issue of the claims in the application.

Respectfully Submitted

By:

Volel Emile
Attorney for Applicants
Registration No. 39,969
(512) 306-7969